

Application Mobile

« Gestion de bibliothèque »

Projet académique - Ingénierie Informatique et Réseaux

Réalisé par :

FAHIM IMANE

OUBAALLA EL MEHDI

ZOUIZZA KHALIL

Supervisé par :

M. NADIRI Abdeljalil

Table des matières

<i>I. Introduction.....</i>	<i>3</i>
<i>II. Objectifs du Projet</i>	<i>3</i>
<i>Gestion des Utilisateurs.....</i>	<i>3</i>
<i>Gestion des Livres.....</i>	<i>3</i>
<i>Interface Utilisateur Réactive</i>	<i>3</i>
<i>III. Technologies Utilisées.....</i>	<i>3</i>
<i>1- Structure de l'Application</i>	<i>4</i>
<i>2- Outils et technologies</i>	<i>7</i>
<i>IV. Réalisation.....</i>	<i>8</i>
<i>V. Conclusion</i>	<i>18</i>

I. Introduction

Le rapport suivant représente une explication du projet « Gestion de bibliothèque » qui est une application mobile intuitive et efficace, permettant aux utilisateurs de gérer les informations relatives aux livres et aux membres d'une bibliothèque. Cette application vise à simplifier les processus de gestion, d'emprunt et de retour de livres, tout en offrant une interface conviviale pour les administrateurs et les utilisateurs.

II. Objectifs du Projet

L'application mobile est conçue pour répondre aux besoins suivants :

Gestion des Utilisateurs

Permettre aux administrateurs de modifier et de supprimer des utilisateurs déjà enregistrés lors de l'interface « Register » selon leurs rôles.

Gestion des Livres

Offrir la possibilité de gérer les livres disponibles, y compris l'ajout de nouveaux titres, la mise à jour des informations existantes et la suppression des livres ainsi que le suivi des emprunts.

Interface Utilisateur Réactive

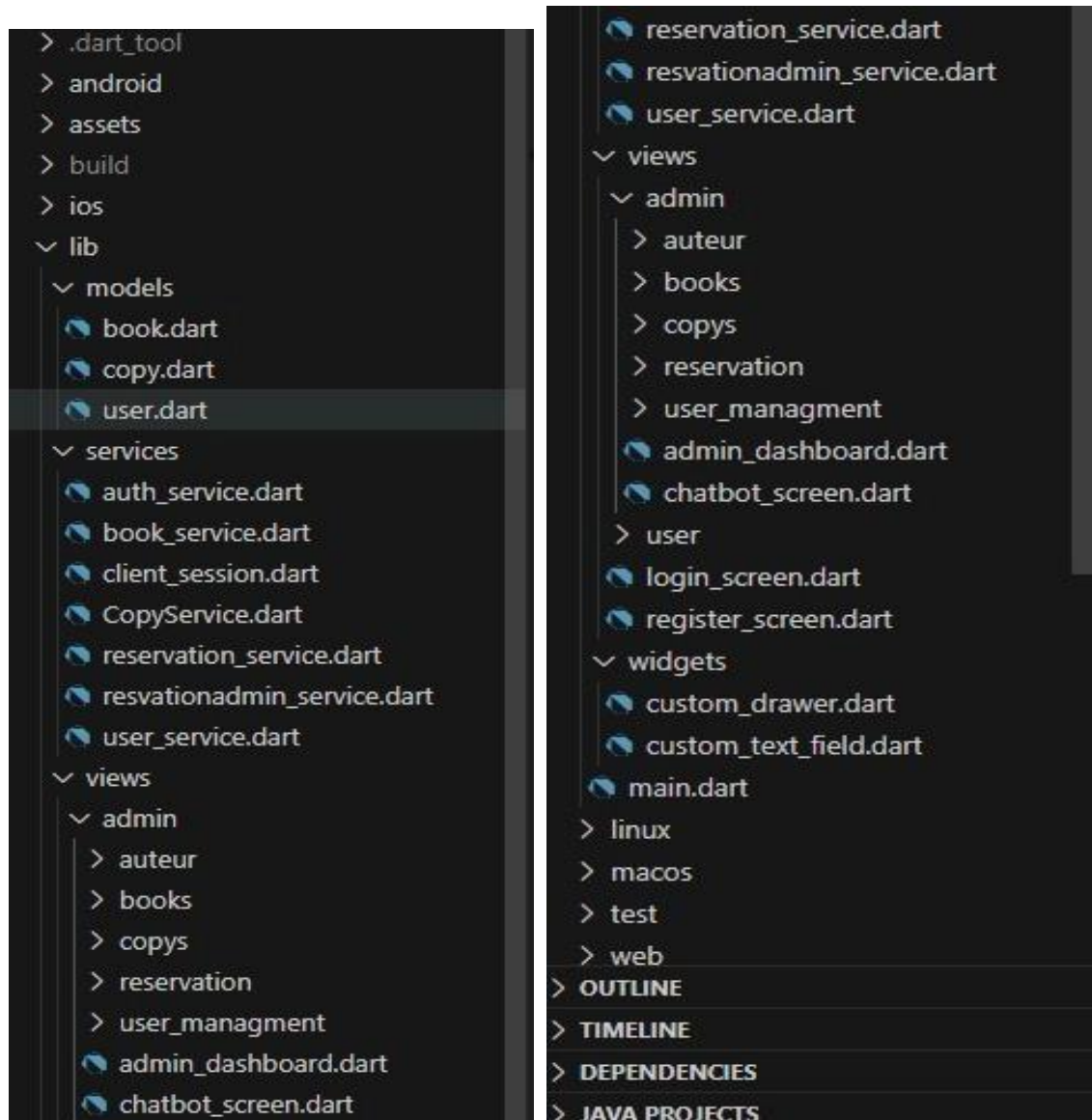
Développer une interface utilisateur fluide et réactive, optimisée pour les appareils mobiles, afin d'améliorer l'expérience utilisateur.

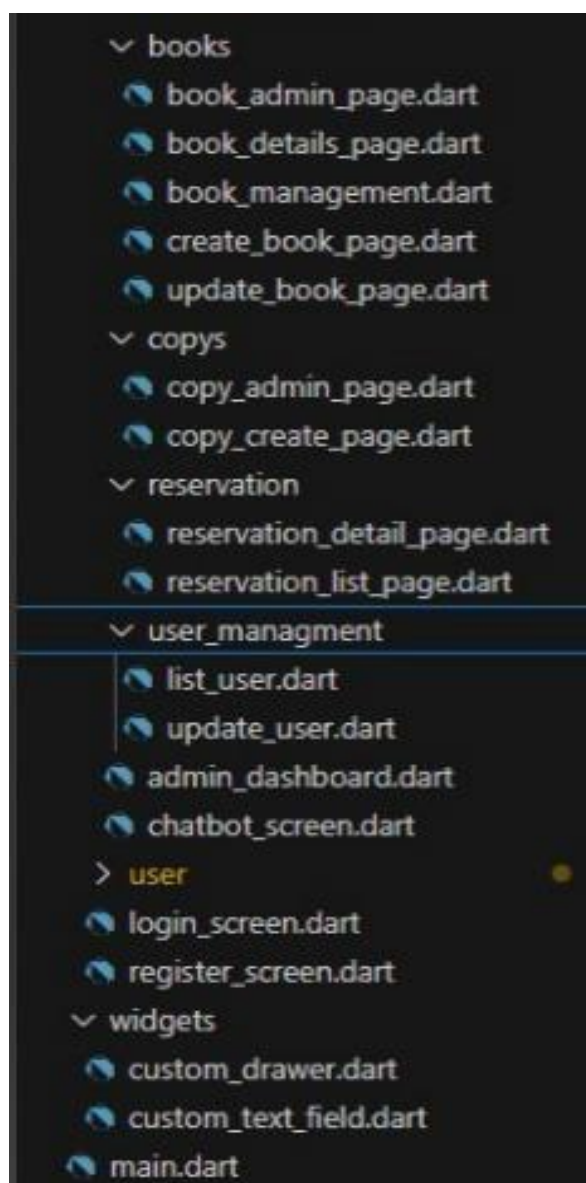
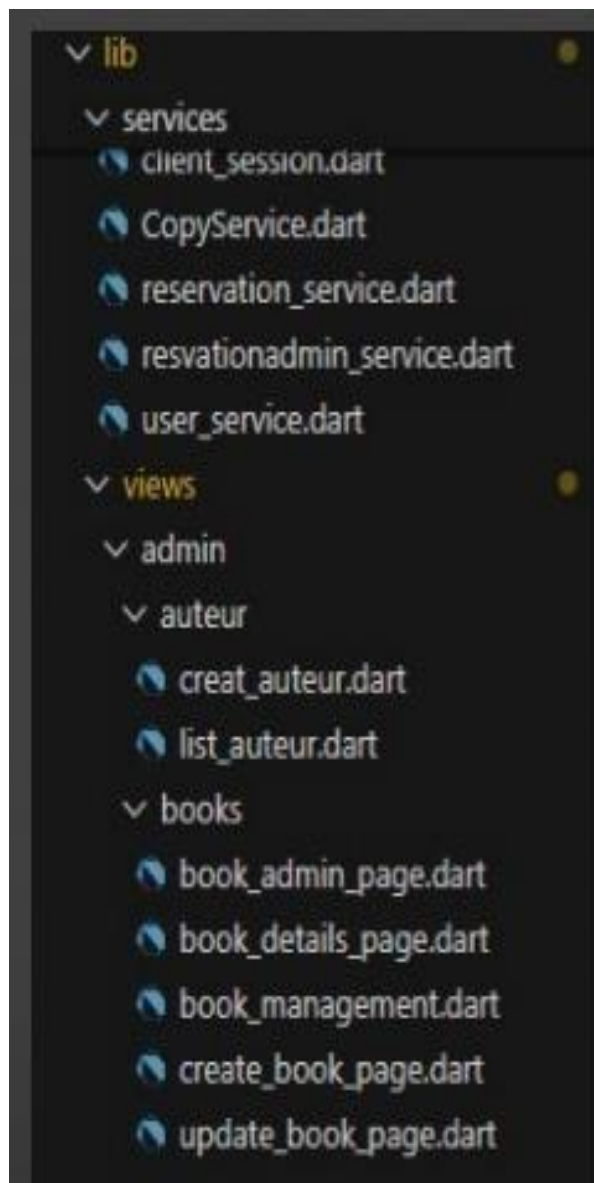
III. Technologies Utilisées

L'application est développée en utilisant Flutter, un Framework moderne qui permet de créer des applications multiplateformes avec une seule base de code.

Le backend développé par Spring est géré par une API RESTful, qui facilite les opérations de création, lecture, mise à jour et suppression (CRUD) des données.

1- Structure de l'Application





a. Modèle-Vue-Contrôleur (MVC)

L'application suit le modèle architectural MVC, qui divise l'application en trois composants principaux :

Modèle :

Représente les données. Dans notre projet, les modèles incluent des classes comme User et Book, qui définissent la structure des données.

Vue :

Gère l'interface utilisateur. Les fichiers de vue, tels que ceux dans le dossier views, contiennent les widgets Flutter qui composent l'interface utilisateur. Par

exemple, list_user.dart et create_book_page.dart définissent comment les utilisateurs et les livres sont affichés et interagissent avec l'utilisateur.

Contrôleur :

Gère la logique de l'application et les interactions entre le modèle et la vue. Les services, comme user_service.dart et book_service.dart, agissent comme des contrôleurs, encapsulant la logique de communication avec l'API backend et la gestion des données. Ils effectuent des requêtes HTTP pour récupérer ou envoyer des données.

Les services incluent :

UserService : Gère les opérations liées aux utilisateurs, telles que la récupération des utilisateurs, la mise à jour des informations et la suppression des utilisateurs.

BookService : Gère les opérations liées aux livres, y compris la création, la mise à jour et la récupération des livres.

AuthService : Gère l'authentification des utilisateurs, y compris la connexion et la gestion des tokens.

b. Sécurité

L'application utilise des tokens JWT (JSON Web Tokens) pour gérer l'authentification des utilisateurs. Les tokens sont stockés dans

SharedPreferences pour une utilisation persistante, et chaque requête vers le backend inclut ce token dans les en-têtes pour assurer la sécurité des opérations.

2- Outils et technologies :

Flutter



Flutter est un framework open-source développé par Google pour créer des applications multiplateformes à partir d'une seule base de code. Il permet de développer des applications pour iOS, Android, le web et le bureau avec une interface utilisateur réactive et performante. Flutter utilise le langage de programmation Dart, qui facilite la création d'interfaces utilisateur modernes et fluides.

Dart



Dart est un langage de programmation orienté objet développé par Google, utilisé principalement pour le développement d'applications Flutter. Il est conçu pour être facile à apprendre et à utiliser, avec une syntaxe claire et des fonctionnalités modernes telles que la gestion des types, la programmation asynchrone et la compilation juste-à-temps (JIT) pour des performances optimales.

Spring Boot



Framework principal pour le développement d'applications Java. Il simplifie la configuration et le déploiement d'applications basées sur Spring.

Maven



Outil de gestion de projet et de construction utilisé pour gérer les dépendances et le cycle de vie du projet.

MySQL



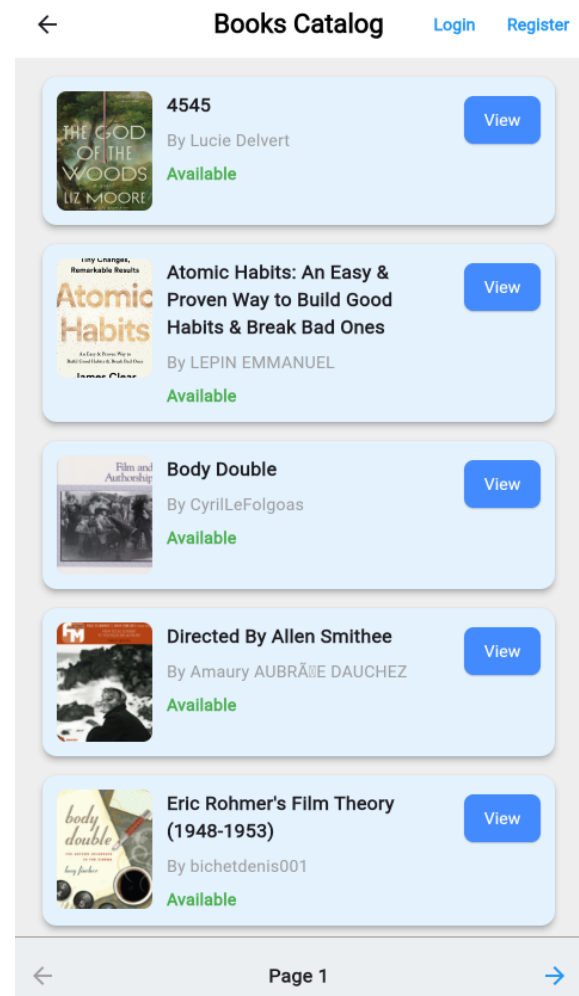
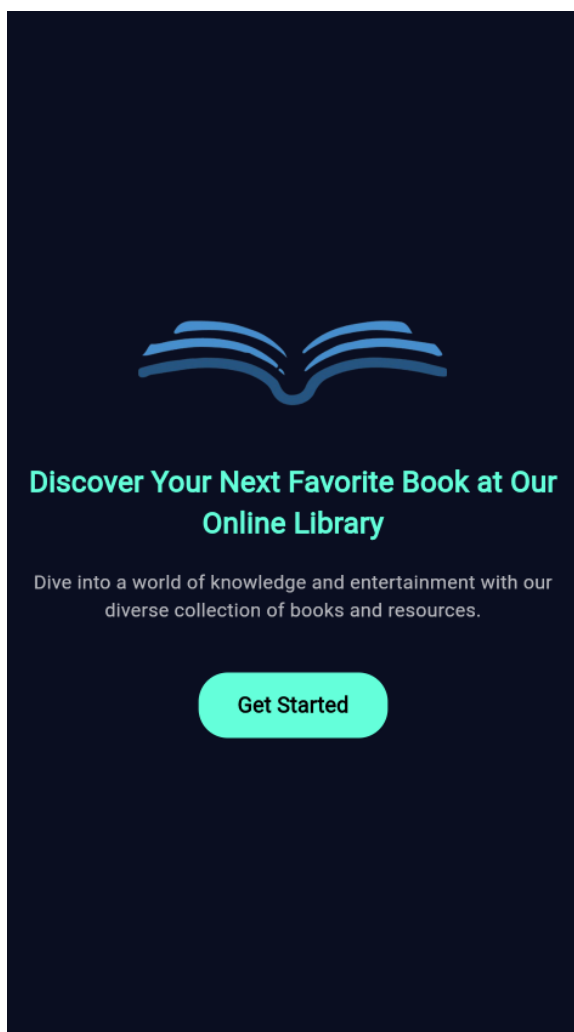
Système de gestion de base de données relationnelle utilisé pour stocker les données de l'application.

IV. Réalisation

Client :

Les captures d'écran suivantes représentent la partie qui s'affiche lorsqu'un client et connecte ainsi que l'ensemble des étapes et interfaces avec lesquelles il interagit pour pouvoir effectuer une réservation d'un livre.

Home page :





Create Account

Fill in the details below to create an account.

Register

Already have an account? [Login](#)



Login

Enter your email and password

Email:

Password:

Login

Don't have an account? [Register](#)

[Forget Password?](#)

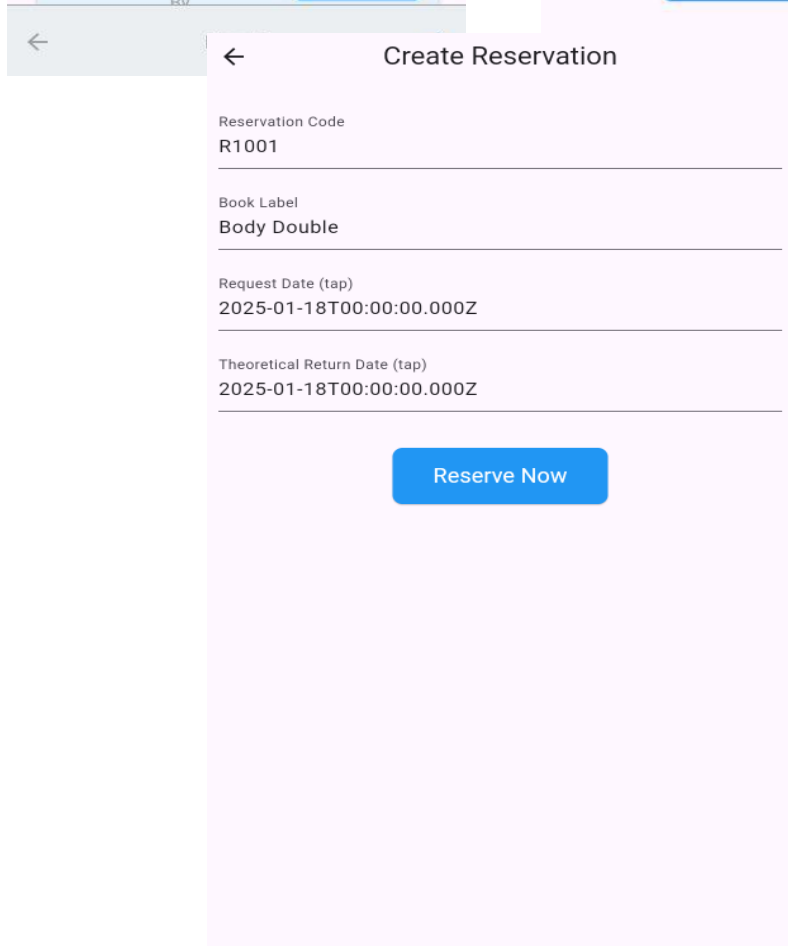
← Email Verification

Verify Your Account

Enter the activation code sent to your email to activate your account.

Verify





← Create Reservation

Reservation Code
Body Double

Request Date (tap)

Theoretical Return Date (tap)

Reserve Now

← Reservation Detail

Reservation Code: anamobile
Book Title: anamobile

Select a Copy

Select Return Date

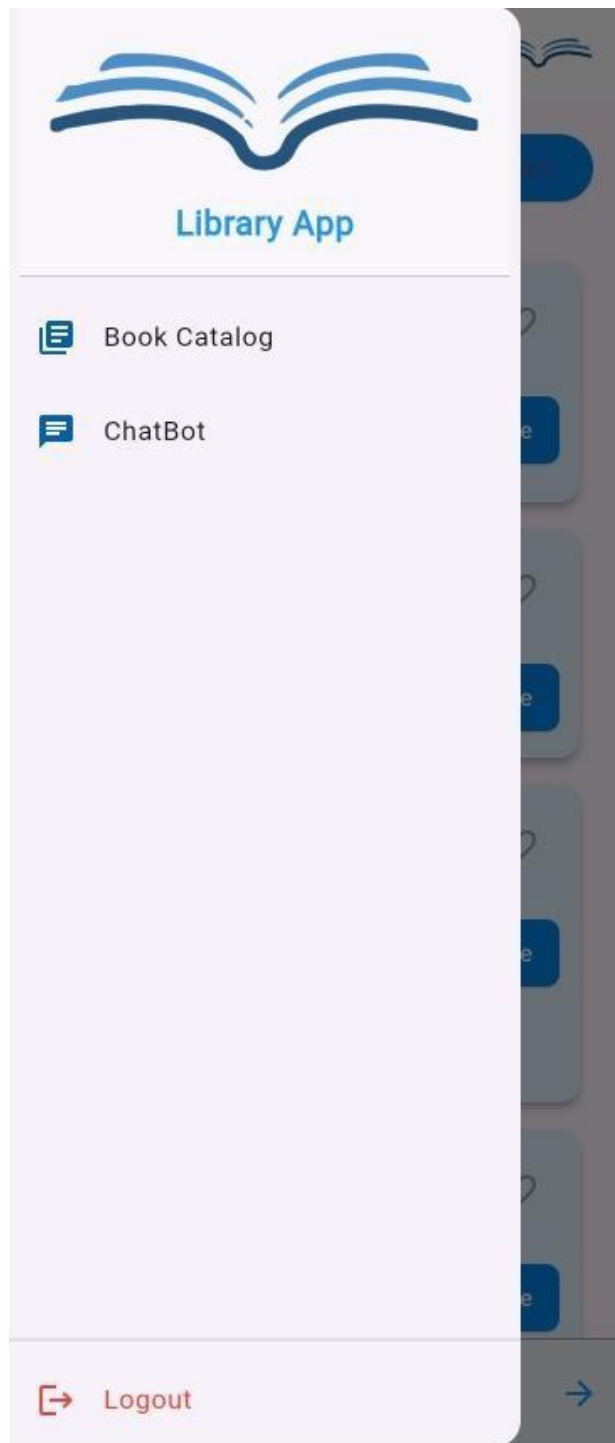
Validate Reservation

← My Reservations

Body Double
Requested on: 01/18/2025 01:00
Status: Completed

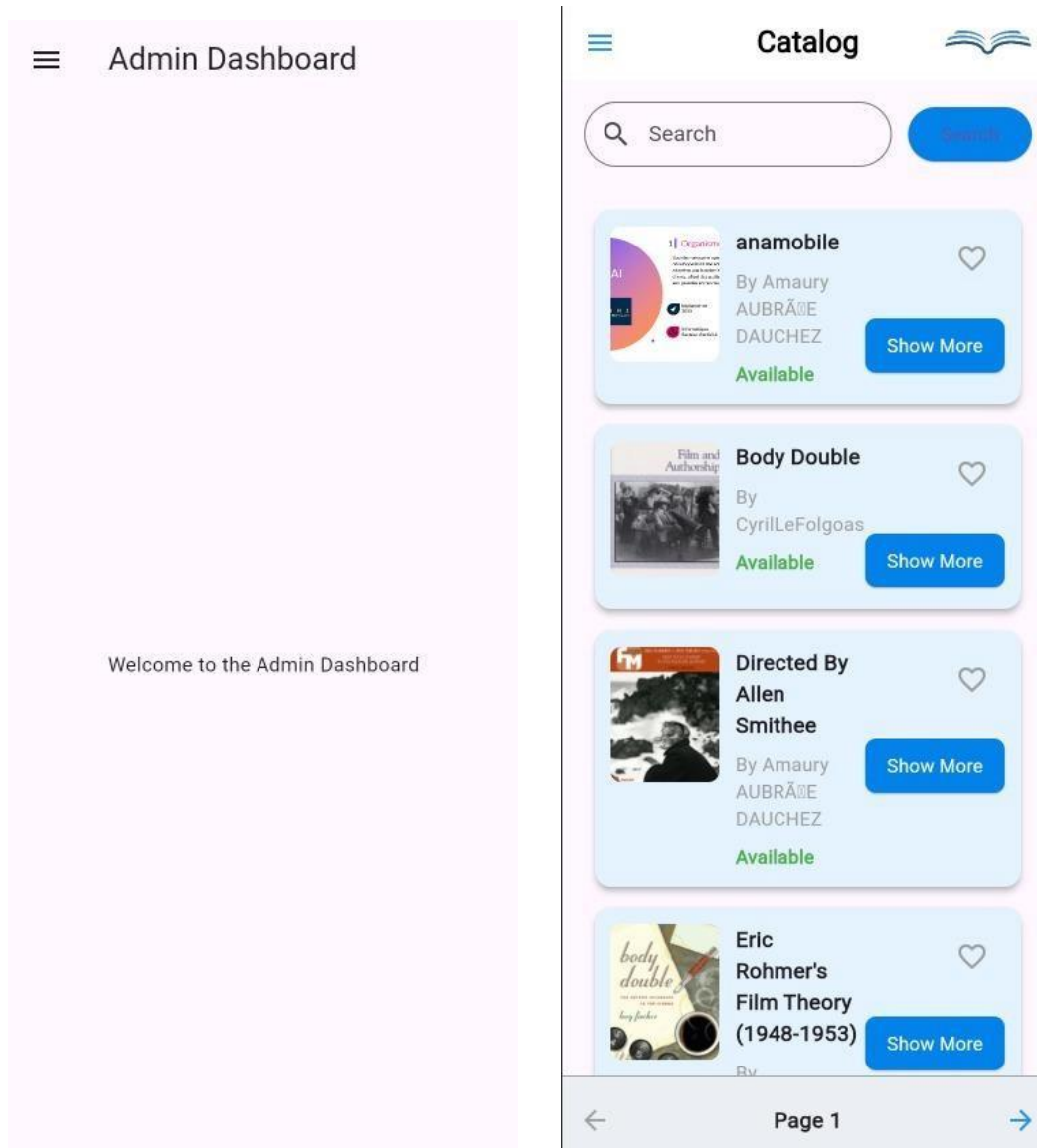
Directed By Allen Smithee
Requested on: 01/18/2025 01:00
Status: Pending

Body Double
Requested on: 01/18/2025 01:00
Status: Completed



Admin :

Les captures d'écran suivantes représentent la partie qui fournit des interfaces conviviales pour l'admin pour qu'il puisse gérer les utilisateurs, les livres, les copies et les réservations, avec une option de chat bot.



←

Manage Copies

Create New Copy

Serial: anamobile
Book Title: anamobile

Serial: anamobile1
Book Title: anamobile

←

Page 1

→

←

Reservations

Code: anamobile
Request Date: 01/14/2025
23:23
Client: client

Affect Copy

Code: yztz
Request Date: 01/09/2025
01:00
Client: client

Affect Copy

Code: me95
Request Date: 01/09/2025
01:00
Client: client

Affect Copy

Code: zyn-1722764291
100fdezdefdefe
Request Date: 01/10/2025
01:00
Client: client

Affect Copy

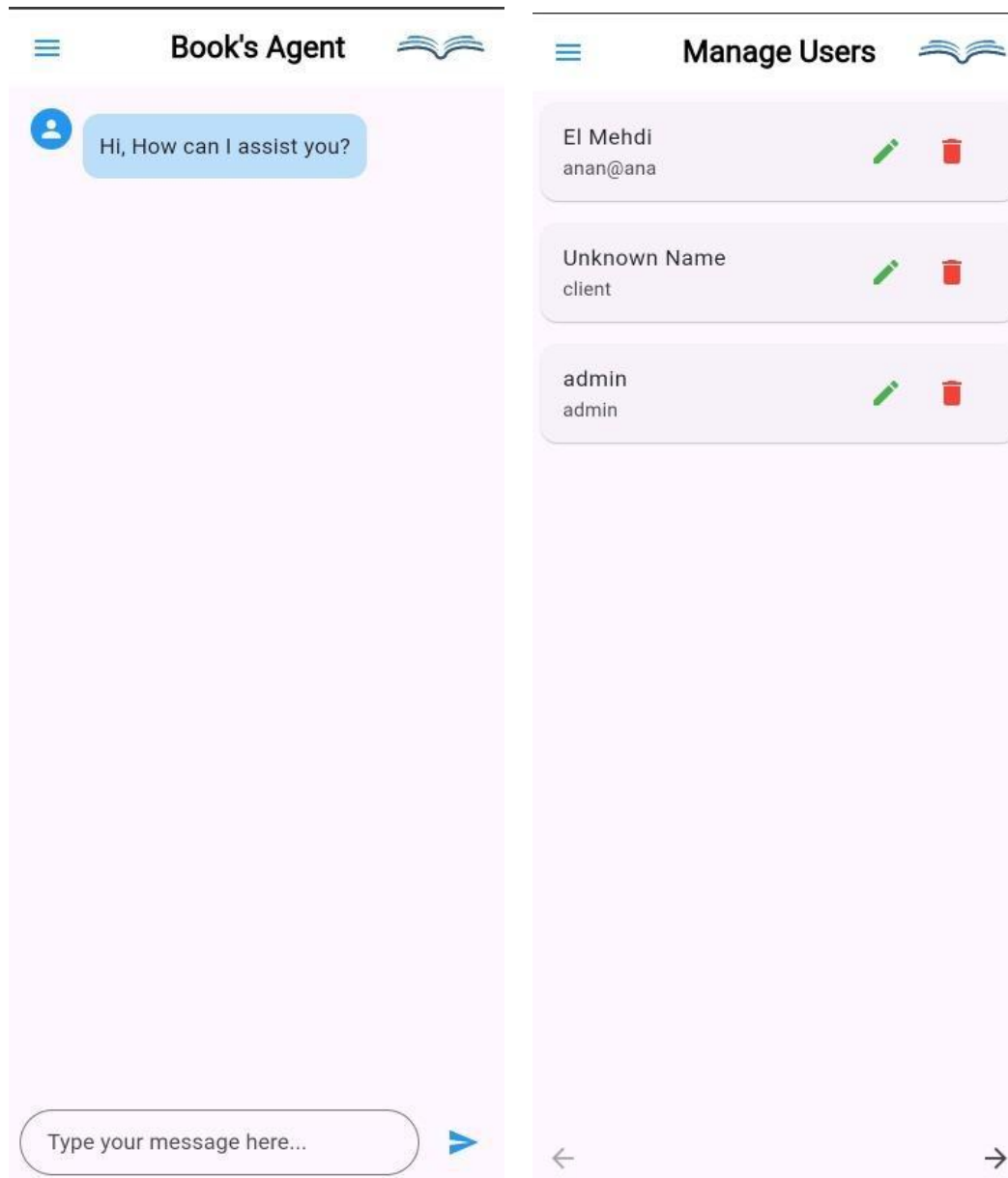
Code: zyn-1722764291
100fdezdgbf
Request Date: 01/07/2025
19:22
Client: client

Affect Copy

←

Page 1

→



←

Update User

Email

admin

Username

admin

Phone

0000000000

First Name

admin

Last Name

admin

ROLE_ADMIN ▾

Update User

☰


Manage Books

📖

🔍 Search

🔍

Available




Body Double

By CyrilLeFolgoas

Available

✎ 🗑

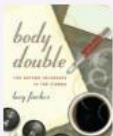


Directed By Allen Smithee

By Amaury AUBRÉE DAUCHEZ

Available

✎ 🗑



Eric Rohmer's Film Theory (1948-1953)

By bichetdenis001

Available

✎ 🗑

+

Film and

←

Page 1

→

← **Create Book (Web)**

Code*

Label*

Title*

Select Edition Date*

Description

Number of Copies*


Available* ☒


Select Author* ▼

Select Category* ▼

Add Copy

Pick Image

← **Update Book** 



Code
Body Double

Label
Body Double

Title
Body Double


Edition Date
01/03/2019 00:59

Description
Body Double explores the myriad ways th

Number of Copies
0

Image URL
<http://localhost:9000/books/21.png>

Update Book

 **Manage Reservations**

Reservation Code: R1005
Book: Body Double
Client: client



Completed ▾


Reservation Code: R1004
Book: Directed By Allen Smithee
Client: client


Pending ▾


Reservation Code: R1001
Book: Body Double
Client: client


Completed ▾


 Page 1 



Library App


 Book Catalog


 Manage Books

 Manage copys

 Manage Reservations

 Manage Users

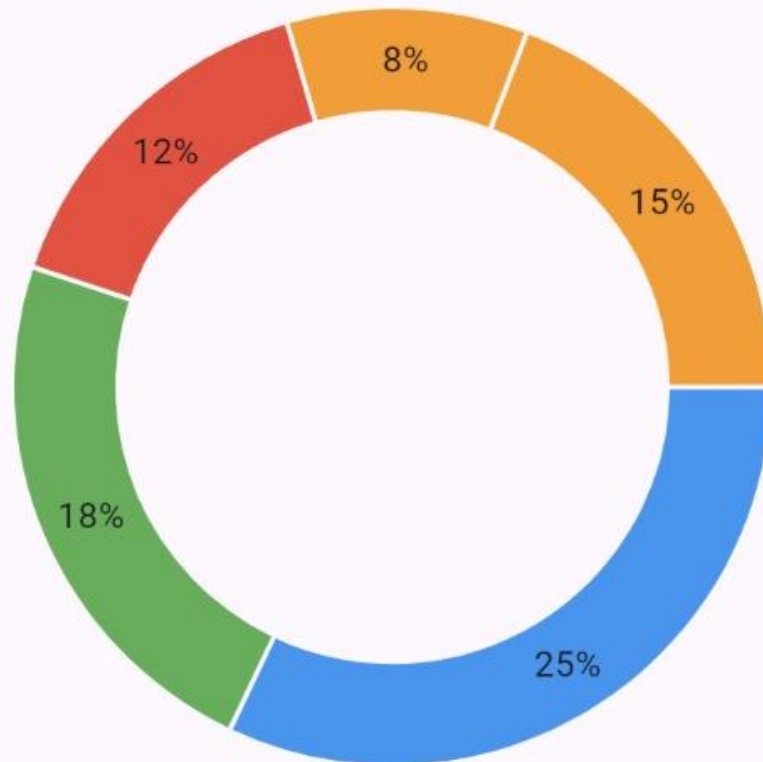
 ChatBot

 Logout

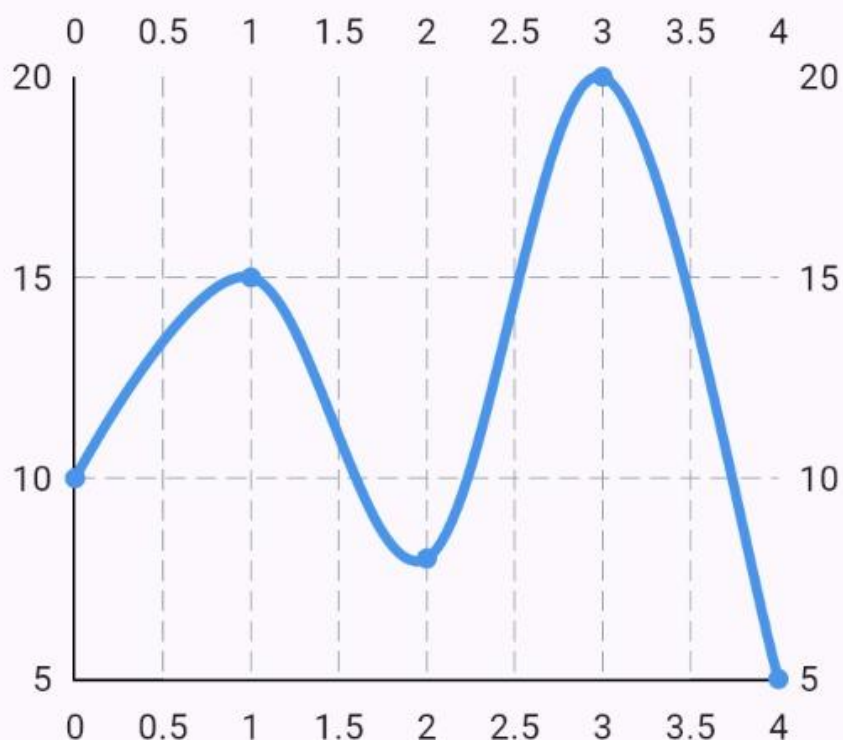
Admin Dashboard



Category Distribution



Author Contributions



V. Conclusion

Le projet de gestion de bibliothèque a abouti à une solution conviviale et robuste pour la gestion des utilisateurs et des livres, en s'appuyant sur une architecture bien définie basée sur le modèle MVC, ce qui facilite le développement et la maintenance. L'équipe a rencontré plusieurs défis, notamment la gestion des types de données lors de la communication avec le backend et l'intégration de mécanismes de sécurité. Malgré ces obstacles, l'application offre des fonctionnalités essentielles, telles que la gestion des utilisateurs et des livres, tout en garantissant une interface utilisateur intuitive. Les perspectives incluent l'ajout de fonctionnalités supplémentaires, comme la gestion des notifications, ainsi que l'optimisation pour d'autres plateformes, ce qui ouvre la voie à des initiatives similaires dans le domaine de la gestion des ressources.